Optimal allocation sequences of two processes sharing a resource

Bruno Gaujal

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE

Optimal Allocation Sequences of Two Processes Sharing a Resource

Bruno Gaujal

N° 2223

Mars 1994

PROGRAMME 1

Architectures parallèles,

bases de données,

réseaux et systèmes distribués

Rapport de recherche

1994

ISSN 0249-6399

Optimal Allocation Sequences of Two Processes Sharing a Resource

Bruno Gaujal *

Programme 1 — Architectures parallèles, bases de données, réseaux
et systèmes distribués

Projet Mistral

Rapport de recherche n° 2223 — Mars 1994 — 21 pages

Abstract: In this paper, we show that the notion of most regular word in the language formed by all of the words containing a fixed number of each letter of an alphabet can be applied to find the optimal resource allocation policy of the generic system composed by two processes sharing a resource. This system will be modeled as a Petri net. This result can be partially generalized to non periodic allocation sequences and non rational frequencies. For N processes sharing a resource, we show that no such optimal sequence exists. In this case, we give an heuristic to find a good allocation sequence.

Key-words: Optimal Scheduling, Petri Nets, Regular Word

(Résumé : tsvp)

This work was done while the author was visiting AT&T Bell Laboratories, Murray Hill, N.J.

*Bruno.Gaujal@sophia.inria.fr

Unité de recherche INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex (France)

Séquence optimale d'allocations pour deux processus partageant une ressource

Résumé : Dans cet article nous montrons que la notion de mot le plus régulier dans le language formé de tous les mots qui contiennent un nombre fixe de chaque lettre de l'alphabet permet de trouver la politique d'allocation optimale d'un système composé de deux processus qui partagent une ressource avec des fréquences imposées; ce système étant modélisé sous forme de résau de Petri. Le résulat peut être partiellement généralisé à des séquences non périodiques ou non rationnelles. Pour N processus partageant la même ressource, une telle séquence optimale n'existe pas et nous donnons alors une heuristique pour trouver une bonne séquence.

Mots-clé : Ordonnancement , réseaux de Petri, mots réguliers

1 Introduction

We consider the generic system composed of two processes are sharing a common resource. Each process goes through a number of activities (operations) in a cyclic manner. There are two types of activities depending on whether or not the common resource is being used. For each process, the set of activities utilizing the common resource is followed by the set of activities not utilizing that resource. Each activity has a fixed duration or temporization. The above system is typical in manufacturing applications. For instance, in a manufacturing work-cell, the two processes could correspond to two workstations while the common resource corresponds to a robot in charge of material handling.

Without any external control, the system follows its natural evolution which merely depends on the internal characteristics of the system (e.g., temporizations of the activities and the interaction between the two processes). This case has already been studied in [Gaujal 93] and the system reaches its natural periodic regime after a transient behaviour.

Here, we impose an external constraint on the system frequency. Tthe ratio of allocations of the resource to each process is fixed and prespecified. Now, given the resource allocation ratio (or frequency), we are interested in finding the optimal resource allocation sequence so that some performance measures, such as, the cycle time to complete N allocations or the idle time of the resource is minimized. To solve this problem, we introduce the notion of "most regular word" in a formal language composed by words where each letter has to appear a given number of times (its "frequency").

In many scheduling problems, the "maximal delay" argument caracterizes the optimal solution. However, when the system that we are dealing with is cyclic (or periodic) then the good notion that has to be applied is rather "regularity". This general principle is once more verified in this case.

In the second section, we introduce the notion of most regular word, in the third section we present some of its basic properties.In section 4, we present our application. In the fifth section we study the system with only two processes. Section 6 shows that with 3 processes, no strongly optimal sequence exists. Section 7 gives some extensions of section 5.

2 Most Regular Words

$A = \{a_1, a_2, \cdots, a_n\}$ is the alphabet. Let w be a word written on the alphabet A. $|w|$ is the total number of letters in w, $|w|_{a_i}$ is the number of a_i's in w.

Let $k = k_1+k_2+\cdots+k_n$. $L(k_1, \cdots, k_n)$ is a language on the alphabet A: $L = \{w s.t. |w|_{a_i} = k_i \quad \forall i\}$ In other words, $L(k_1, \cdots, k_n)$ is the set of all the words of length k containing

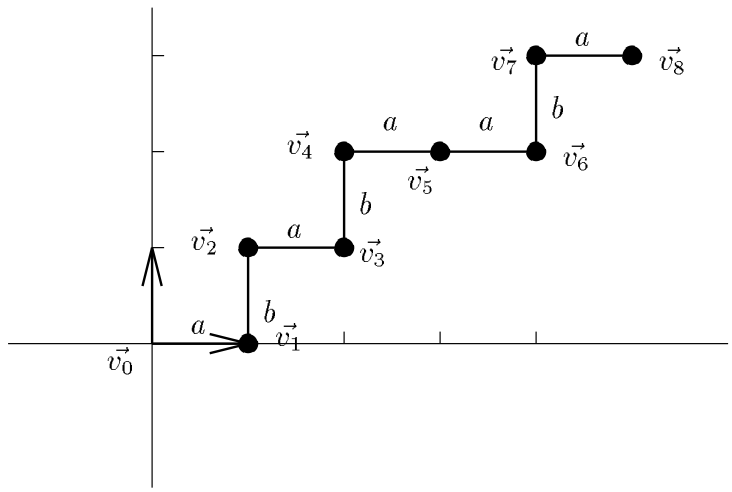

Figure 1: Mapping in $I\!R^2$ of the word *ababaaba*.

k_1 a_1's, k_2 a_2's,$\cdots$, k_n a_n's. k_i is called the *frequency* of the letter a_i in the language $L(k_1, \cdots, k_n)$.

2.1 Mapping on $I\!R^n$

We denote by :

$$\begin{array}{rcl} \vec{e}_1 & = & (1, 0, 0, \cdots, 0) \\ \vec{e}_2 & = & (0, 1, 0, \cdots, 0) \\ \vdots & \vdots & \ddots \\ \vec{e}_n & = & (0, 0, \cdots, 0, 1) \end{array}$$

the natural base of $I\!R^n$.

We can map $L(k_1, \cdots, k_n)$ onto a path in R^n in the following way. Each letter of A is associated to one dimension: $t(a_i) = \vec{e}_i$.

A word $w = w_1 w_2 \cdots w_n$ is associated to the sequence of vectors: $v = (\vec{v}_0, \vec{v}_1, \cdots, \vec{v}_n)$ where $\vec{v}_i = t(w_1) + t(w_2) + \cdots + t(w_i)$.

We set $t(w) = v$ (see figure 1).

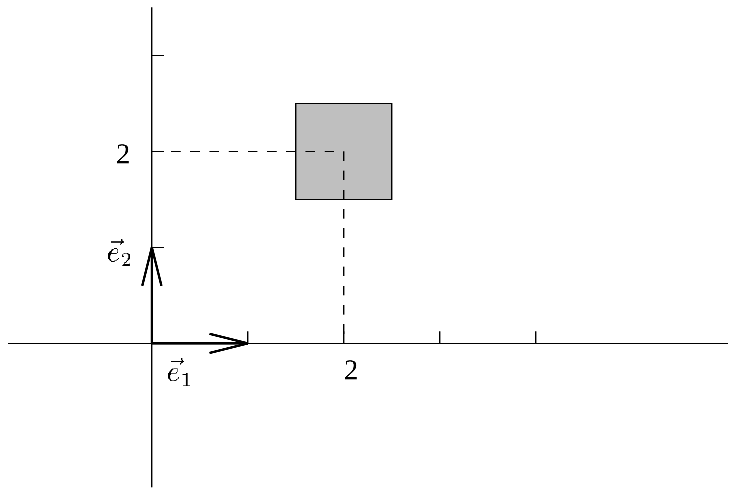

Figure 2: A Voronoi cell in dimension 2.

Remark 1: $\forall w \in L,\ t(w)_0 = \vec{v}_0 = (0, 0, \cdots, 0)$ and $t(w)_n = \vec{v}_n = (k_1, k_2, \cdots, k_n)$.

Remark 2: $\forall w \in L, t(w)$ is contained in the n-dimensional cube between the points $(0, 0, \ldots, 0)$ and $(k_1, k_2, \cdots, k_n)$.

2.2 The Most Regular Word

This notion is well known in formal language theory, see [Rauzy 84] and discrete geometry (discrete lines), see [Berstel 90]. Here we provide *ad hoc* notations and properties of these sequences and we do not pretend to generality or exhaustivity.

We define the Voronoi cells centered at the integer points in $I\!R^n$ by: If $\vec{c}$ is a vector in $I\!N^n$, $V(\vec{c})$ is the set all the point in $I\!R^n$ which are closer to $\vec{c}$ than to any other point in $I\!N^n$. $V(\vec{c} = \{\vec{x} \in I\!R^n : d_\infty(\vec{x}, \vec{c}) \leq 1/2\}$. See figure 2 which depicts the Voronoi cell of center $(2, 2)$ in $I\!R^2$.

We define the *center diagonal* $D(k_1, \cdots, k_n)$ to be the line segment in $I\!R^n$ between the points of coordinates $= (0, 0, \cdots 0)$ and $(k_1, k_2, \cdots, k_n))$. The *most regular* word in the language $L(k_1, \cdots, k_n)$ is the word $r(k_1, \cdots, k_n)$ such that the sequence $t(r)$ is "as close as possible" to the center diagonal $D(k_1, \cdots, k_n)$.

More precisely, we partition $I\!R^n$ in the Voronoi cells centered at the integer points. We consider the cells that intersect D. We call them *diagonal* cells.

If one point $\vec{p}$ in D belongs to three cells or more, we remove some of these cells. All these cells have a face of dimension $n-k < n-1$ in common to which $\vec{p}$ belongs. The diagonal cell that contain a segment of D before $\vec{p}$ is kept as the "first" cell V_1 containing $\vec{p}$. We denote by $\vec{p}_1$ its center. The diagonal cell that contains a segment of D after the point $\vec{p}$ is denoted V_2 and its center $\vec{p}_2$.

$\vec{p}_2 = \vec{p}_1 + \lambda_1\vec{e}_1 + \cdots + \lambda_n\vec{e}_n$. Since these 2 cells have a face in common, $\lambda_i \in \{0,1\}$ for all i. We can rewrite the previous equation: $\vec{p}_2 = \vec{p}_1 + \vec{e}_{\sigma(1)} + \cdots + \vec{e}_{\sigma(k)}$, σ being an increasing function. From all the cells that contain $\vec{p}$, we only keep the cells with centers $\vec{p}_1, \vec{p}_1 + \vec{e}_{\sigma(1)}, \vec{p}_1 + \vec{e}_{\sigma(1)} + \vec{e}_{\sigma(2)}, \cdots, \vec{p}_2 = \vec{p}_1 + \vec{e}_{\sigma(1)} + \cdots + \vec{e}_{\sigma(k)}$.

Here, we somehow give priority to a_1 over $a_2 \cdots$... over a_n for the words in $L(k_1, \cdots, k_n)$.

The first diagonal cell is the cell $V(\vec{c}_0)$ with $\vec{c}_0$ of coordinates $(0, 0, \cdots, 0)$ because the point $\vec{c}_0$ belongs to D. Since two consecutive diagonal cells ($V(\vec{c}_j)$ and $V(\vec{c}_{j+1})$) share a face, say the ith face of $V(\vec{c}_j)$, to go from the center of one cell ($V(\vec{c}_j)$) to the center of the following one, we add the vector e_i and we know that its center is $\vec{c}_{j+1} = \vec{c}_j + \vec{e}_i$. Finally, the center of the last diagonal cell is the point $(k_1, k_2, \cdots, k_n)$. The total number of cells is k since we go from the point $(0, 0, \cdots, 0)$ to the point $(k_1, \cdots, k_n)$ only by adding unit vectors in all the n directions. Furthermore, the sequence $(\vec{c}_0, \cdots, \vec{c}_k)$ is the image of a word in $L(k_1, \cdots, k_n)$ by t. This word is called the *most regular* word in $L(k_1, \cdots, k_n)$ and is denoted by $r(k_1, \cdots, k_n)$.

Another way to see the most regular word can be derived using only discrete points on D. For all $i \leq k$ let $\vec{d}_i$ be the vector with coordinates $i/k(k_1, \cdots, k_n)$. $\vec{d}_i$ belongs to the hyperplane P_i defined by the equation $x_1 + x_2 + \cdots + x_n = i$. We can see that $\vec{c}_i$ is the closer point in P_i to $\vec{d}_i$ with integer coordinates. Note that this implies $d(\vec{d}_i, \vec{c}_i) \leq \sqrt{n}/2$.

This discrete version allows one to note that the most regular word is the word r in $L(k_1, \cdots, k_n)$ where the "deficit" or the "surplus" in any letter of any prefix of r is as small as possible.

2.3 Examples

1. $A = \{a, b\}$, $k = 6, k_1 = 4, k_2 = 2$

$$\begin{aligned} L(4,2) = \; & \{aaaabb, aaabab, aaabba, aabaab, aababa, aabbaa, abaaab, \\ & abaaba, \\ & ababaa, abbaaa, baaaab, baaaba, baabaa, babaaa, bbaaaa\}. \end{aligned}$$

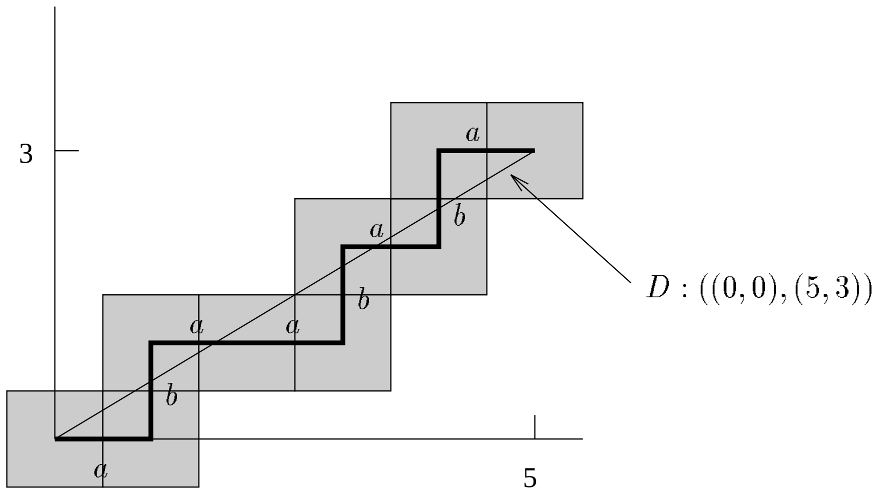

Figure 3: Construction of the most regular word *abaababa* in the language $L(5,3)$. The diagonal voronoi cells are filled with a shade of gray and the most regular word is written along its mapping onto IR^2.

The most regular word in $L(4,2)$ is *abaaba*.

2. $A = \{a, b\}$, $k = 8, k_1 = 5, k_2 = 3$

   We give in figure 3 the construction of $r(5,3) = abaababa$.

3. $A = \{a, b, c\}$, $k = 3m, k_1 = k_2 = k_3 = m$.

   The most regular word in $L(m, m, m)$ is: $(abc)^m$.

   We see through these three examples that the most regular word corresponds to what our intuition would see as the most "even" or "balanced" words in the language. However this intuition does not give any clue when the parameters are more general.

4. $A = \{a, b, c\}$, $k_1 = 10, k_2 = 3, k_3 = 7$. The most regular word is $r(10,3,7) = acabcaacacbacaacbaca$. To construct this word we used a property that is given in the next section. We construct a regular word of dimension n from the n regular words in dimension $n - 1$.

3 Preliminaries Properties

We present a few properties of most regular words. Some of them will be useful in the following applications, some of them may have an interest by their own. However, we don't pretend to exhaustivity.

Property 1 *If r is the most regular word in $L(k_1, \cdots, k_n)$, then the projection along one axis (say e_1) of r in $I\!R^n$ gives the most regular sequence in $L(k_2, \cdots, k_n)$.*

Proof The diagonal in $I\!R^n$:

$$D_n = ((0, \cdots, 0), (k_1, \cdots, k_n)),$$

is projected into the diagonal in $I\!R^{n-1}$:

$$D_{n-1} = ((0, \cdots, 0), (k_2, \cdots, k_n)).$$

The diagonal Voronoi cells in $I\!R^n$ are projected into the diagonal Voronoi cells in $I\!R^{n-1}$. Since the most regular word is formed by the center of the diagonal cells, the projected word is the most regular word in the projected space. ■

In other words, if r is the most regular word in $L(k_1, k_2, \cdots, k_n)$ on the alphabet $\{a_1, a_2, \cdots, a_n\}$, and if we remove all the a_1's in r, we obtain the most regular word in $L(k_2, \cdots, k_n)$ on the alphabet $\{a_2, \cdots, a_n\}$.

The following property is the reciprocal.

Property 2 *If $r_1, r_2, \cdots, r_n$ are the most regular words in the sets $L(k_2, \cdots, k_n)$, $L(k_1, k_3, \cdots, k_n), \cdots$, $L(k_1,\ cdots, k_{n-1})$ respectively, then the word r which projections along each dimension are $r_1, r_2, \cdots, r_n$ is the most regular word in $L(k_1, \cdots, k_n)$.*

Proof Consider the Voronoi cells in $I\!R^n$ whose projection on each subspace are the Voronoi diagonal cells in the $n-1$dimensional subspace. Since the projection of the diagonal $((0, \cdots, 0), (k_1, \cdots, k_n))$ is the diagonal in each subspace, the diagonal is contained in those Voronoi cells which in turn are the diagonal cells in $I\!R^n$. Therefore, their centers give the most regular word in the n dimensional set $L(k_1, \cdots, k_n)$. ■

This property implies that the most regular word in dimension n can be constructed from the n most regular words in dimension $n-1$. The example 4 has been constructed using this method.

Property 3 *Let $d = \gcd(k_1, \cdots, k_n)$. The most regular word in $L(k_1, \cdots, k_n)$ cut after the k/d-th letter is the most regular word in $L(k_1/d, \cdots, k_n/d)$.*

Proof the diagonal D passes through the integer point $(k_1/d, \cdots, k_n/d)$. and this portion of D is the diagonal for the words in $L(k_1/d, \cdots, k_n/d)$. ■

Property 4 *If r contains two consecutive a_j's, then $k_j > k_i$ for any i.*

Proof Let us consider the projection in the 2-dimensional space (e_i, e_j). r contains two consecutive a_j means that the diagonal passes through three Voronoi cells with centers $(x, y), (x, y+1), (x, y+2)$. Therefore its slope is bigger than 1. Since its slope is k_j/k_i, this ends the proof. ■

An easy corollary of this property is that, r cannot contain two consecutive a_i's and two consecutive a_j's for any $i \neq j$.

3.1 The special case on the alphabet with two letters

We consider the alphabet $A = \{a, b\}$, and the language $L(k_a, k_b)$. This case will be very important in the following. The properties shown in this part will be for the most part technical lemmas useful in the following.

All the following properties will be illustrated using the example with the parameters $k_a = 3, k_b = 7$.

In this part, we will use a different representation of the language $L(k_a, k_b)$. Indeed, it can also be coded using the following observation. Any word in $L(k_a, k_b)$ on the alphabet $A = a, b$ can be written: $w = b^{w_0}ab^{w_1}ab^{w_2}\cdots ab^{w_{k_a}}$ where $w_0 + w_1 + w_2 + \cdots + w_{k_a} = k_b$ and $w_i \geq 0, \ \forall i$.

We define a *shift* to be a one-cycle permutation and we are also interested in the set $G(k_a, k_b)$ of all the words in $L(k_a, k_b)$ which are shifts of the most regular word, $r = b^{r_0}ab^{r_1}ab^{r_2}\cdots ab^{r_{k_a}}$. The set $G(k_a, k_b)$ is called the regular subset of $L(k_a, k_b)$.

$G(k_a, k_b)$ can be described by the k_a-tuple $[r_1, \ldots, r_{k_a} + r_0]$. Indeed, this tuple represents the word $w = ab^{r_1}ab^{r_2}\cdots ab^{r_{k_a}+r_0}$ which is a shift of r. And clearly, the description of one word in $G(k_a, k_b)$ defines the whole set completely.

In $L(3, 7)$, the most regular word is $r = babbabbbab$ and $G(3, 7)$ can be defined by the tuple $[2, 3, 2]$.

let $[m_1, \ldots, m_{k_a}]$ (resp. $[M_1, \ldots, M_{k_a}]$) be the minimal (resp. maximal) tuple in $G(k_a, k_b)$ in terms of partial sums. This means that for any $j < k_a$ and for any word $w = (w_1, \ldots, w_{k_a})$ in $G(k_a, k_b)$, $\sum_{i \leq j} m_i \leq \sum_{i \leq j} w_i$ (resp. $\sum_{i \leq j} M_i \geq \sum_{i \leq j} w_i$). This words m and M are both obtained by shifting r. If $t(r) = (\vec{c}_0, \cdots, \vec{c}_k)$ where $\vec{c}_i$ is a center of the i-th diagonal Voronoi cell, consider the point $\vec{c}_J$ such that $d(\vec{c}_J, \vec{d}_J)$ is maximal on all the points $\vec{c}_i$ *above* the diagonal D. Now $t(m) = (\vec{0}, \vec{c}_{J+1} - \vec{c}_J, \cdots, \vec{c}_k - \vec{c}_J, \vec{c}_k + \vec{c}_1 - \vec{c}_J, \vec{c}_k + \vec{c}_2 - \vec{c}_J, \cdots, \vec{c}_k + \vec{c}_{J-1} - \vec{c}_J)$ rewritten as $(\vec{u}_0, \vec{u}_1, \cdots, \vec{u}_k)$.

Similarly, consider the point $\vec{c}_I$ such that $d(\vec{c}_I, \vec{d}_I)$ is maximal on all the points $\vec{c}_i$ *below* the diagonal D. Now $t(M) = (\vec{0}, \vec{c}_{I+1} - \vec{c}_I, \cdots, \vec{c}_k - \vec{c}_I, \vec{c}_k + \vec{c}_1 - \vec{c}_I, \vec{c}_k + \vec{c}_2 - \vec{c}_I, \cdots, \vec{c}_k + \vec{c}_{I-1} - \vec{c}_I)$ rewritten as $(\vec{U}_0, \vec{U}_1, \cdots, \vec{U}_k)$.

Note that since $d(\vec{d}_i, \vec{c}_i) < \sqrt{2}/2$, for all i, $d(\vec{d}_i, \vec{u}_i) < \sqrt{2}$. Similarly, $d(\vec{d}_i, \vec{U}_i) < \sqrt{2}$.

In $G(3, 7)$, the minimal tuple is [2,2,3] and the maximal one is [3,2,2]. In figure 4, the most regular word is displayed twice: *babbabbbabbabbabbbab* to allow representation of shifts on the same figure. The word m corresponds to the sequence (BD) i.e. *abbabbabbb*. If the origin is put in B, the line (BD) becomes the center diagonal and one can notice that m is below (BD). The word M is the word between A and C, i.e. *bbbabbabba*. If the origin is moved to A, the line (AC) becomes the diagonal and M is above it.

Property 5 $m_i = \lfloor i \frac{k_b}{k_a} \rfloor - \lfloor (i-1) \frac{k_b}{k_a} \rfloor$. $M_i = \lceil i \frac{k_b}{k_a} \rceil - \lceil (i-1) \frac{k_b}{k_a} \rceil$.

Proof We define the words $t(k_a, k_b)$ and $T(k_a, k_b)$ by $t(k_a, k_b)_i = (\lfloor i \frac{k_b}{k_a} \rfloor - \lfloor (i-1) \frac{k_b}{k_a} \rfloor)$ and $T(k_a, k_b)_i = (\lceil i \frac{k_b}{k_a} \rceil - \lceil (i-1) \frac{k_b}{k_a} \rceil)$. First, note that the two sequences m and $t(k_a, k_b)$ are below the diagonal D. Then, note that $t(k_a, k_b)$ is above m since any sequence which is not below t is not below D. Finally, note that if $t(k_a, k_b)$ is strictly above m then for some index i, $d(\vec{d}_i, t(m_i)) \geq \sqrt{2}$ which contradicts the definition of m. The proof of $M = T(k_a, k_b)$ is similar. ■

In our example, we can verify that :

$$\begin{aligned}
\lfloor 1\frac{7}{3} \rfloor &= 2 \\
\lfloor 2\frac{7}{3} \rfloor - \lfloor 1\frac{7}{3} \rfloor &= 2 \\
\lfloor 3\frac{7}{3} \rfloor - \lfloor 2\frac{7}{3} \rfloor &= 3
\end{aligned}$$

$$\lceil 1\frac{7}{3} \rceil = 3$$

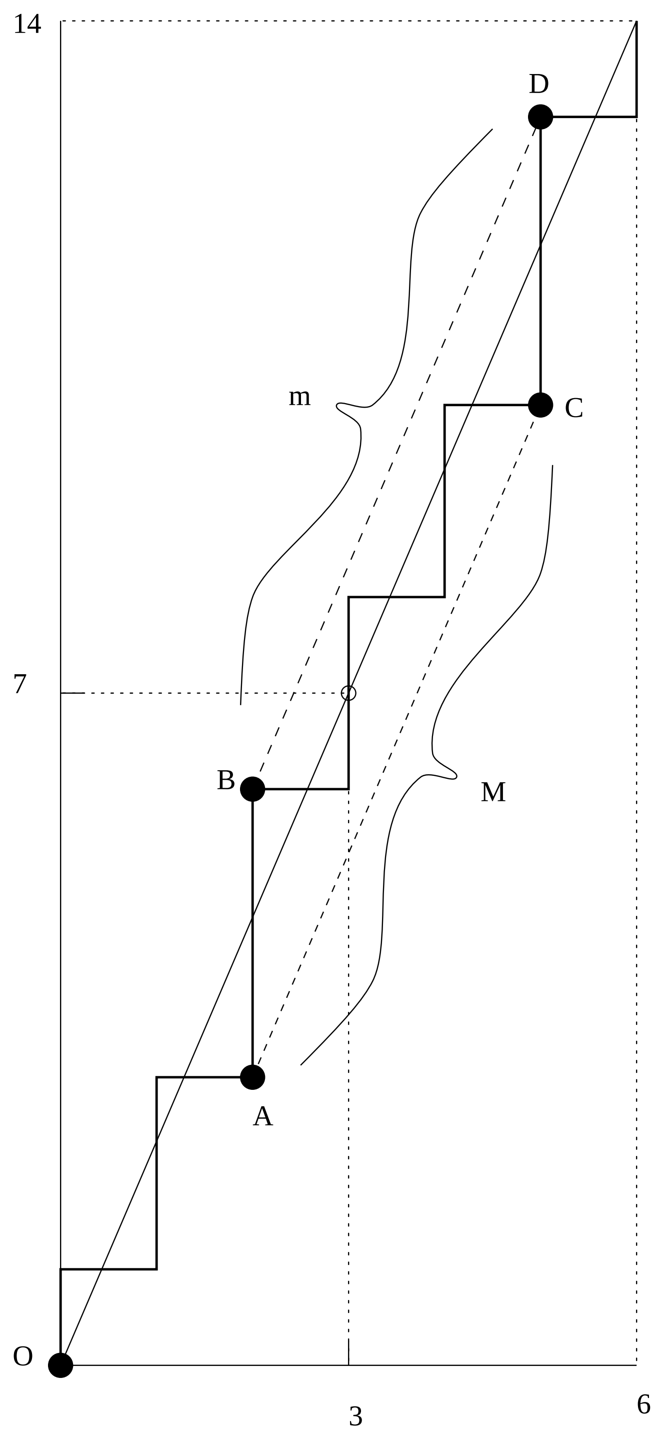

Figure 4: Representation of the most regular word in $L(3,7)$ displayed twice (from $(0,0)$ to $(6,14)$), the minimal word and the maximal word in $G(3,7)$.

$$
\begin{aligned}
\lceil 2\frac{7}{3}\rceil - \lceil 1\frac{7}{3}\rceil &= 2 \\
\lceil 3\frac{7}{3}\rceil - \lceil 2\frac{7}{3}\rceil &= 2
\end{aligned}
$$

Property 6 *Assume $k_b > k_a$. If m is the minimal word in $L(k_a, k_b)$, then $m' = m - 1$ is the minimal word in $L(k_a, k_b - k_a)$.*

Proof

$$
\begin{aligned}
m'_i &= \lfloor i\frac{k_b - k_a}{k_a}\rfloor - \lfloor (i-1)\frac{k_b - k_a}{k_a}\rfloor \\
&= \lfloor i(\frac{k_b}{k_a} - 1)\rfloor - \lfloor (i-1)(\frac{k_b}{k_a} - 1)\rfloor \\
&= \lfloor i\frac{k_b}{k_a}\rfloor - \lfloor (i-1)\frac{k_b}{k_a}\rfloor - 1.
\end{aligned}
$$

If $k_a > k_b$ then $m_i \in \{0, 1\}$ for all $i < k_a$ because of property 4.

Property 7 *If $k_a > k_b$, if m is the minimal word in $L(k_a, k_b)$, then $M' = 1 - m$ is the maximal word in $L(k_a, k_a - k_b)$.*

Proof

$$
\begin{aligned}
M'_i &= 1 - m_i \\
&= 1 - \lfloor i\frac{k_b}{k_a}\rfloor + \lfloor (i-1)\frac{k_b}{k_a}\rfloor \\
&= i - \lfloor i\frac{k_b}{k_a}\rfloor - \left((i-1) - \lfloor (i-1)\frac{k_b}{k_a}\rfloor\right) \\
&= \lceil i - i\frac{k_b}{k_a}\rceil - \lceil (i-1) - (i-1)\frac{k_b}{k_a}\rceil \\
&= \lceil i\frac{k_a - k_b}{k_a}\rceil - \lceil (i-1)\frac{k_a - k_b}{k_a}\rceil.
\end{aligned}
$$

4 One Application: Scheduling of Two Processes Sharing a Resource

We will apply the notion of most regular words to a scheduling problem. However we believe that this notion can be used in many areas including other scheduling problems, like in ([Hajek 85]).

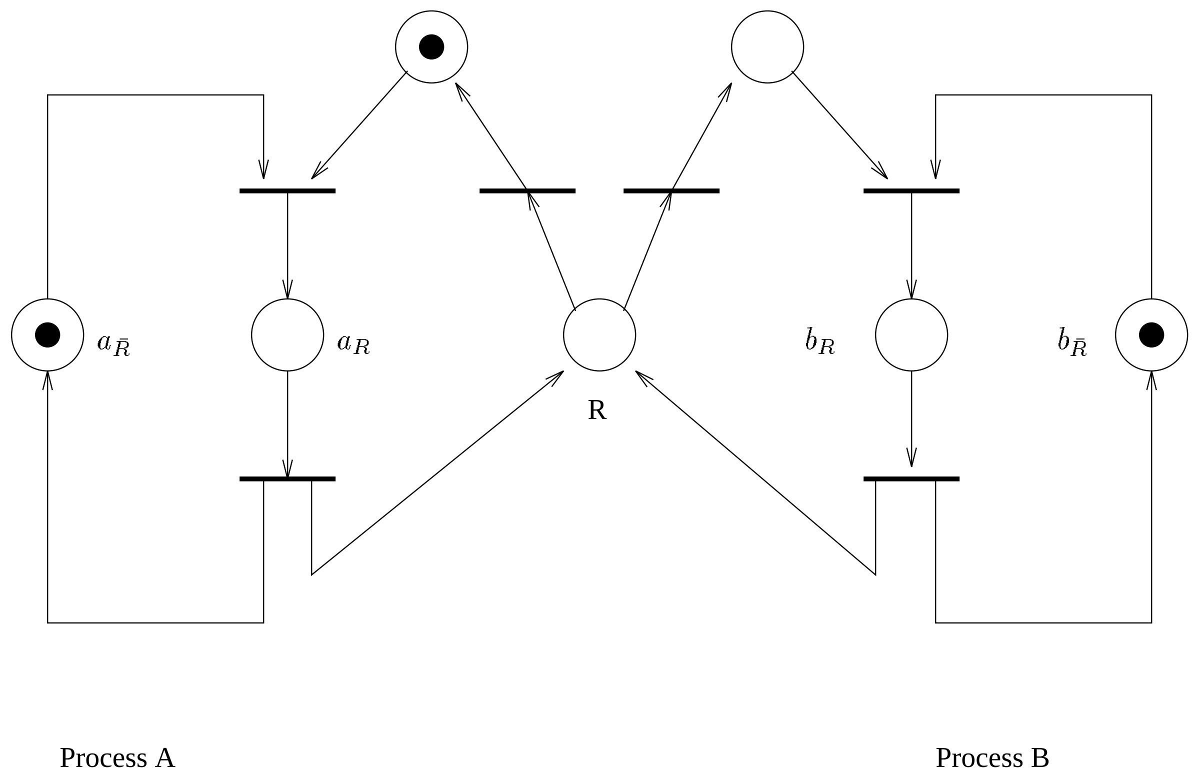

Figure 5: Free choice net modeling the system S. The resource is available if there is a token in place R. This token can be routed to any process regardless of the state of the system.

4.1 Description of the Problem

In this section, we consider the resource allocation in the system (S) composed by n cyclic processes $A_1, \cdots, A_n$ using the resource in mutual exclusion. Each process (A_i) is cyclic and performs two kinds of activities during its cycle: an activity that does not use the resource during a total time of $a_i^{\bar{R}}$ alternating with an activity that uses the resource during a_i^R. Furthermore, we have an external control on the resource: when it is available, we can select the process that gets it. Figure 5 is a Petri Net model of this system in the case $n = 2$. Processes A and B share the resource R. The free choice place R gives total control on the resource allocation regardless of the state of the processes.

Our purpose is to find the resource allocation sequence minimizing the idle time of the resource given the following constraint on the frequency of resource allocations.

We are interested in the class of periodic scheduling. The ratio of allocations between the n processes is fixed and must be met after k allocations. We denote by k_i the number of times that the resource must be given to process A_i within k allocations. The set of all the valid allocation sequences is denoted $V(k_1, \cdots, k_n)$.

We remark that minimizing the idle time of the resource is equivalent to minimizing the total time required to go through k allocations. Indeed, if we denote by M the total time of running the system for k allocations, the idle time of the resource during this period is $M - \sum_i k_i a_i^R$.

One can notice that a valid allocation sequence of length k corresponds to a word in $L(k_1, \cdots, k_n)$ on the alphabet $\{A_1, \cdots, A_n\}$ with $k_1 + \cdots + k_n = k$. We can also rewrite this as $V(k_1, \cdots, k_n) = L(k_1, \cdots, k_n)^\omega$. In the following, we restrict ourselves to allocation sequences which are periodic of period k. We call such sequences k-periodic allocation sequences. Such an allocation sequence is determined by its first k values and we abusively write: $w = (w_1, \cdots, w_k)$ so that we can also say that $w \in L(k_1, \cdots, k_n)$. However, we will see later that the restriction to periodic sequences is not a strong one.

5 Two Processes Sharing a Resource

We first study the case where $n = 2$. In this case, we adopt the following notations. The two processes are denoted A and B with respective timings $(a_R, a_{\bar{R}})$ and $(b_R, b_{\bar{R}})$, and the resource has to be given k_A times to A and k_B times to B within $k = k_A + k_B$ allocations.

As the problem is symmetric, we will assume that $a_R + a_{\bar{R}} \geq b_R + b_{\bar{R}}$. In the following, we will "desymmetrize" the problem and this inequality will be used at the very end of the proof of the theorem.

Once a k-periodic allocation sequence is chosen, the system (S, w) can be modeled as a Decision Free Petri Net (or Marked Graph) denoted $MG(S, w)$. See figure 6. Now, the problem can be set in the following from: find an optimal sequence, that is a sequence s such that the cycle time of the system $MG(S, s)$ as minimal among all the systems with sequences in $L(k_A, k_B)$. It has already been shown in [Chaouiya 92] that the "worst" sequence, i.e. the sequence that maximizes the idle time of the resource is the sequence $A \cdots AB \cdots B$. this intuitively corresponds the a serialization of the 2 processes, first A works alone and then B works alone. The following theorem determines the optimal sequence.

Theorem 1 *The optimal k-periodic allocation sequence is the most regular word in $L(k_A, k_B)$.*

Note that the optimal allocation sequence does not depend on the temporizations of the system. In this sense, we say that r a strongly optimal sequence since it is optimal for

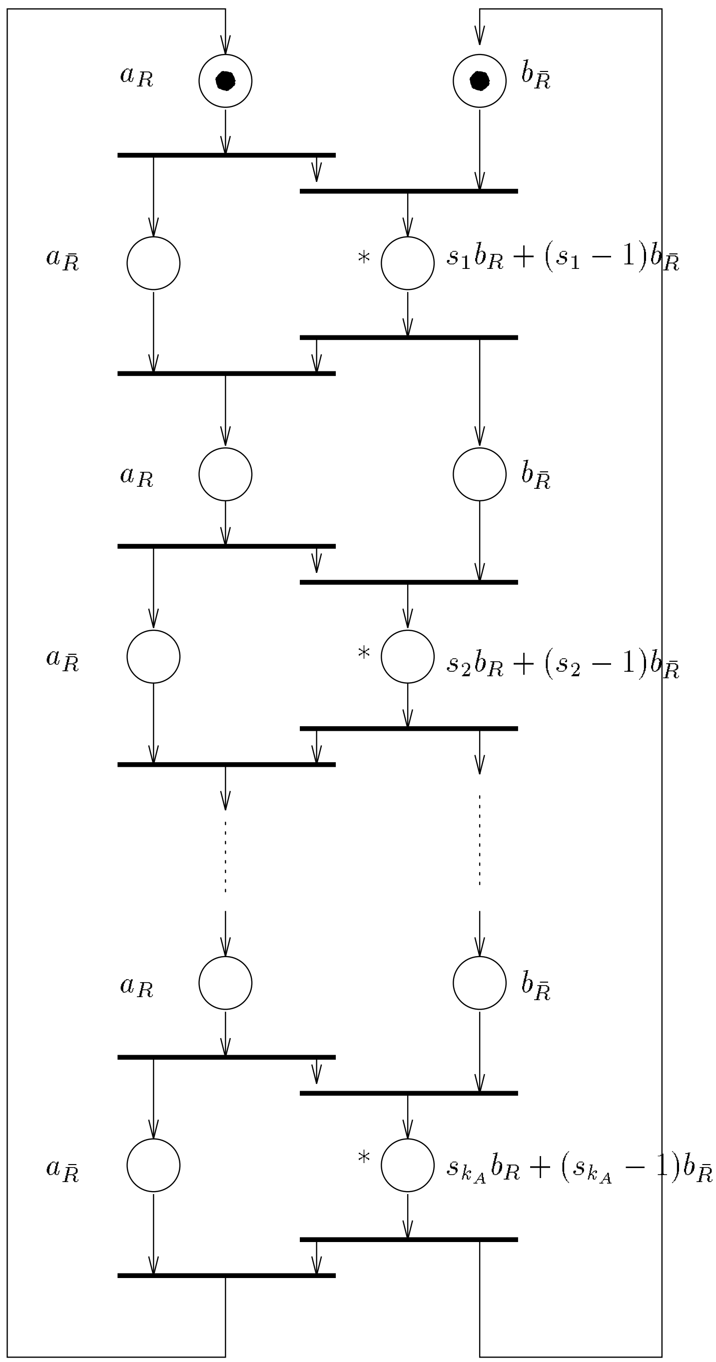

Figure 6: marked graph $MG(S, w)$, where $w = AB^{s_1} \cdots AB^{s_{k_A}}$
* The places with weight $s_i\beta_1 + (s_i - 1)\beta_2$ result from the aggregation of $2s_i - 1$ places with holding times $\beta_1, \beta_2, \cdots, \beta_1$.

all possible temporizations of the 2 processes. This result has already been proved in the restrictive case: k_B is a multiple of k_A in [Gaujal 93]. However, their method cannot be extended in the general case. The proof given here is radically different.

5.1 Temporization of a Circuit

Let s be an allocation sequence. $s = B^{s_0}AB^{s_1}A\cdots AB^{s_{k_A}}$. The system can be modeled by the marked graph $MG(S,s)$. Note that if s' is an allocation sequence which is a shift of s, then $MG(S,s') = MG(S,s)$, but with a different initial marking.

As the system with one allocation sequence is modeled by a marked graph, (see figure 6), we are interested in the cycle time of these nets. We define the temporization of a circuit in a timed marked graph by the sum of the temporization of all the places belonging to this circuit. It has been proved that the cycle time of the marked graph is equal to the temporization of its slowest circuit ([Baccelli 92]). For more details and further references on Marked Graphs, see [Murata 84] for example. This result implies that the cycle time of the system does not depend on the initial marking. Therefore, in our problem, the cycle time of $MG(S,s')$ is the same as the cycle time of $MG(S,s)$. Any shift of the sequence s will have the same performance and we can choose one representative in the class of s that we denote by $(s_1, s_2, \cdots, s_{k_A})$ in the following.

In our case, let C_A (C_B) be the circuit passing through all the places belonging to process A (B) in $MG(S,s)$, see figure 6. If C is a circuit in $MG(S,s)$, it passes through some places in C_A and some places in C_B. If C passes through the aggregated places in C_B with weights $s_{i_1}(b_R + b_{\bar{R}}) - b_{\bar{R}}, \cdots, s_{i_l}(b_R + b_{\bar{R}}) - b_{\bar{R}}$ and p is the number of times that C passes from a place in C_B to one in C_A, then the temporization of C can be put in the from : $t(C) = (s_{i_1} + \cdots + s_{i_l})(b_R + b_{\bar{R}}) + (k_A - l)(\alpha_1 + a_{\bar{R}}) + p(a_R - b_{\bar{R}})$.

We introduce new objects. A vector e will always denote an element of $\{0,1\}^{k_A}$, an *interval* is a vector I in $\{0,1\}^{k_A}$ of the form: $(0,\cdots,0,1,\cdots,1,0,\cdots,0)$. In other words, there exists two numbers i and j such that $\{e_i, e_{(i+1)\mathrm{mod}k_A}, e_{(i+2)\mathrm{mod}k_A}, \cdots, e_{(i+j)\mathrm{mod}k_A}\}$ are all equal to 1 and the remainings all equal to 0.

We say that e is p-split if e can be decomposed as a sum of p intervals.

$E(p,l) = \{e \in \{0,1\}^{k_A}$ s.t. $1.e = l$ and e is p-split$\}$.

In other words, $E(p,l)$ is the set of all the $(0,1)$-sequences of length k_A containing l 1's distributed in at most p intervals. For example if $k_A = 13$, $e = (1,1,1,0,0,1,0,0,0,1,1,0,1) \in E(3,7)$.

Thus, we can re-write $t(C)$ as: $t(C) = (s.e)(b_R + b_{\bar{R}}) + (k_A - l)(\alpha_1 + a_{\bar{R}}) + p(a_R - b_{\bar{R}})$ for some $e \in E(p,l)$.

The maximum cycle C_s in $MG(S,s)$ has the temporization $t(C_s) = F(p_s, l_s, s)(b_R + b_{\bar{R}}) + (k_A - l_s)(a_R + a_{\bar{R}}) + p_s(a_R - b_{\bar{R}})$.

where

$$F(p,l,s) = \max_{e \in E(p,l)} s.e$$

and (p_s, l_s) is such that $t(C_s)$ is maximum.

A vector e in $E(p,l)$ that verifies $s.e = F(p,l,s)$ is called an argmax of F for s and will be denoted by $e(p,l,s)$.

As an example, let $p = 3, l = 6$. Consider the vector $s = (\underline{1,2,1},0,0,\underline{1,1},1,0,0,0,\underline{2},0,0)$. We get an argmax (corresponding to the underlined elements in s):
$e(3,6,s) = (1,1,1,0,0,1,1,0,0,0,0,1,0,0)$ and $F(3,6,s) = 8$.

5.2 Minimization of the Function F

In this section, we will try to study the function F.

First, note that $F(.,.,s)$ is increasing in l and increasing in p. This comes directly from the definition of $E(p,l)$. We can also derive from the definition of $E(p,l)$ that if y is a shift of x then $F(p,l,y) = F(p,l,x)$.

The reader can also note that $F(1,1,x) = \max_i(x_i)$ and that more generally, $F(p,p,x)$ is the sum all the p biggest elements in x. This remark allows one to deduce the fact that for all $w \in L(k_A,k_B)$, $F(p,p,w) \geq F(p,p,r)$. Is not difficult either to see that $F(p,l,w) \geq F(p,l,r) = k_B$ if $p > k_B$. However, we will prove that this inequality holds for more general cases.

Lemma 1 *If* (p,l) *verifies*

$$\min_{w \in L(k_A,k_B)} F(p,l,w) > \min_{w \in L(k_A,k_B)} F(p,l-1,w) \tag{1}$$

then

$$\min_{w \in L(k_A,k_B)} F(p,l,w) = F(p,l,r).$$

The function $F(p,l,.)$ is convex, for it is the maximum of linear functions. Its minimum in $I\!R^{k_A}$ is clearly at the point $(\frac{k_B}{k_A},\cdots,\frac{k_B}{k_A})$ and is equal to $l\frac{k_B}{k_A}$. This point corresponds to the sequence (D_i) of points on the diagonal $((0,0),(k_A,k_B))$ introduced in section 2. The lemma says that the minimum of $F(p,l,.)$ in $I\!N^{k_A}$ is achieved at the point r which is the closest integer point to the solution in $I\!R^{k_A}$. Furthermore, we believe that this result holds for all the couples (p,l) but we have not been able to show it. However, to prove theorem 1, this partial result will be sufficient. The cases where condition (1) is not satisfied are somehow degenerated cases.

Proof The proof holds in several steps:

1 This first step is used only if $k_B \geq k_A$. We show that if $k_B \geq k_A$, we can write $k_B = ak_A + r$, $r < k_A$, and $\min_{w\in L(k_A,k_B)} F(p,l,w) = al + \min_{w\in L(k_A,r)} F(p,l,w)$.

Proof:

$(\geq)$ Let $s=(s_1,\cdots,s_{k_A})$ verify $\min_{w\in L(k_A,r)} F(p,l,w) = F(p,l,s)$.
We construct $s' = (s_1+a,\cdots,s_{k_A}+a)$. $s' \in L(k_A,k_B)$ and $F(p,l,s') = la + \min_{w\in L(k_A,r)} F(p,l,w)$.

$(\leq)$ Let $t=(t_1,\cdots,t_{k_A})$ verify $\min_{w\in L(k_A,k_B)} F(p,l,w) = F(p,l,t)$.
We define the sequence t' by: $t' = (t_1-a,\cdots,t_{k_A}-a)$. One can see that $t' \in L(k_A,r)$ and $F(p,l,t') = \min_{w\in L(k_A,k_B)} F(p,l,w) - la$. ∎

Thus, we can focus only on the case $k_B < k_A$ (thanks to property 6).

2 In this step, we show that there exists a sequence $s=(s_1,\cdots,s_{k_A})$ that minimizes F, i.e. that verifies $\min_{w\in L(k_A,k_B)} F(p,l,w) = F(p,l,s)$ with the property that all the s_i's are positive.

Proof:

Suppose $s=(s_1,\cdots,s_{k_A})$ is a minimal sequence that contains negative values. We construct s' by:

if $s_{i-1} < 0$ and $s_i \geq 0$ then $s'_i = s_i + s_{i-1}$,
if $s_{i-1} < 0$ and $s_i < 0$ then $s'_i = s_{i-1}$,
if $s_{i-1} \geq 0$ and $s_i \geq 0$ then $s'_i = s_i$,
if $s_{i-1} \geq 0$ and $s_i < 0$ then $s'_i = 0$.

One can verify easily that $s' \in L(k_A,k_B)$. Let $I' = (e_d,\cdots,e_f)$ be any interval in $e' = e(p,l,s')$ and J be the interval $(1_{d-1},\cdots,1_{f-1})$. If $s_{d-1} \geq 0$ and $s_f \geq 0$ then $I'.s = I'.s'$
If $s_{d-1} \geq 0$ and $s_f < 0$ then $J.s = I'.s' + s_{d-1} \geq I'.s'$
If $s_{d-1} < 0$ and $s_f \geq 0$ then $I'.s = I'.s' - s_{d-1} \geq I'.s'$

If $s_{d-1} < 0$ and $s_f < 0$ then $J.s = I'.s'$.
In all the cases, we can find an interval I such that $I.s \geq I'.s'$.

As for any other interval K' in e', the corresponding K does not overlap I, The union all the K form an element $e$in $E(p,l)$ and $es \geq e's'$. Therefore, $F(p,l,s) \geq F(p,l,s')$ which implies that s' is minimal if s is minimal.

If s' contains some negative numbers, we repeat the previous operation until we find a minimal positive sequence. ■

3 Here, we show that there exists a sequence $s = (s_1, \cdots, s_{k_A})$ that verifies

$$\min_{w \in L(k_A,k_B)} F(p,l,w) = F(p,l,s),$$

with all the s_i's in $\{0,1\}$.

Proof:

The proof is very similar to the previous one. Suppose $s = (s_1, \cdots, s_{k_A})$ is a minimal positive sequence that contains values bigger than 2. We construct s' by:

if $s_{i-1} \geq 2$ and $s_i < 2$ then $s'_i = s_i + s_{i-1} - 1$,
if $s_{i-1} \geq 2$ and $s_i \geq 2$ then $s'_i = s_{i-1}$,
if $s_{i-1} < 2$ and $s_i < 2$ then $s'_i = s_i$,
if $s_{i-1} < 2$ and $s_i \geq 2$ then $s'_i = 1$.

One can verify easily that $s' \in L(k_A, k_B)$. Let $e' = e(p,l,s')$. Let $I' = (e_d, \cdots, e_f)$ be any interval in e' and J be the interval $(1_{d-1}, \cdots, 1_{f-1})$. If $s_{d-1} < 2$ and $s_f < 2$ then $I'.s = I'.s'$
If $s_{d-1} < 2$ and $s_f \geq 2$ then $I'.s = I.s' + s_{d-1} - 1 \geq I'.s'$
If $s_{d-1} \geq 2$ and $s_f < 2$ then $J.s = I'.s + s_{d-1} - s_f \geq I'.s + s_{d-1} - 1 = I'.s'$
If $s_{d-1} \geq 2$ and $s_f \geq 2$ then $J.s = I'.s'$.
In all the cases, we can find an interval I such that $I.s \geq I'.s'$.

As for any other interval K' in e', the corresponding K does not overlap I, The union all the K form an element e in $E(p,l)$ and $es \geq e's'$. Therefore, $F(p,l,s) \geq F(p,l,s')$ which implies that s' is minimal.

If s' contains some numbers bigger than 2, we repeat the previous operation until we find a minimal sequence composed of zeros and ones only. We call such a sequence a $\{0,1\}$-sequence. ■

4 At this stage of the proof, we know that the minimum of F is achieved by a $\{0,1\}$-sequence. If $k_B > k_A/2$, we can solve the dual problem with the parameters: $k'_B = k_A - k_B$ $k'_A = k_A$ $l' = k_A - l$, where $k'_B \leq k'_A$. If s' is the optimal sequence of

this dual problem, then $s = 1 - s'$ is the optimal sequence for the original problem. Thanks to property 7, we can focus on the case $k_B \leq k_A/2$. ∎

This implies that in the following we are interested only in sequences where for the number of 1's is bigger than the number of 0's.

5 So far, we have not used the fact that (p, l) verify the condition (1). We show that this condition implies the fact that there exists a minimal sequence s for $F(p, l, .)$ that does not contains a zero at the extremity of one interval of any of its $e(p, l, s)$.

Proof:

We prove this result by contradiction. We make the assumption that for all minimal s, There exist an $e(p, l, s)$ that does contain a zero, Then, take any minimal sequence s_l. We get $e(p, l, s_l).s_l = e's_l$, where e' is the same as $e(p, l, s_l)$ with the extremity of its interval corresponding to a zero of s_l removed. $e' \in E(p, l-1)$. Let s'_{l-1} be a minimal {0,1}-sequence for $F(p, l-1, .)$, then $e(p, l, s'_{l-1}).s'_{l-1} \geq e(p, l, s_l).s_l = e'.s_l \geq e(p, l-1, s'_{l-1}).s'_{l-1}$. Since $e(p, l, s'_{l-1}).s'_{l-1} \leq e(p, l-1, s'_{l-1}).s_{l-1} + 1$ for s_{l-1} is a {0,1}-sequence, one of the 2 previous inequality must be an equality.

- If $e(p, l, s'_{l-1}).s'_{l-1} = e(p, l, s_l).s_l$ then s'_{l-1} is minimal for $F(p, l, .)$ and therefore the assumption applies to it. We can construct $e'' \in E(p, l-1)$ with $e(p, l, s'_{l-1}).s'_{l-1} = e''.s'_{l-1} \geq e(p, l-1, s'_{l-1}).s'_{l-1}$. By definition of $e(p, l-1, s'_{l-1})$, this means that $e''.s'_{l-1} \geq e(p, l-1, s'_{l-1}).s'_{l-1}$. Therefore we get the equality $F(p, l, s_l) = F(p, l-1, s_{l-1})$.
- If $e'.s_l = e(p, l-1, s'_{l-1}).s'_{l-1}$, we get directly the equality $F(p, l, s_l) = F(p, l-1, s_{l-1})$. ∎

6 As we are restricted to the case $k_B < k_A/2$, we show that there exists an optimal sequence $s = (s_1, \cdots, s_{k_A})$ with s_i's in $\{0, 1\}$ and if $s_i = 1$ then $s_{i+1} = 0$.

Proof:

Let $s = (s_1, \cdots, s_{k_A})$ be a minimal sequence. If s contains the sub-sequence 11, then it certainly contains the sub-sequence 011. We transform it into 101 to get s'. If no argmax $e' = e(p, l, s,)$ contains this first 1, $e's \geq e's'$ and s' is minimal. If one argmax e' for s' contains this first 1, then it must contain all 3 elements because e' is not an arg max of s for it contains a zero at one extremity as it is shown in point 5 of the proof. Therefore, $e's' = e's \leq e(p, l, s)s$ We repeat this transformation until the sequence has the [10]-property, i.e. any 1 is followed by a 0. ∎

Note that r, the most regular sequence has this [10]-property.

7 We show that if $p < k_B$ and $l > p$, solving the problem with the parameters (k_A, k_B, p, l) is equivalent than solving the problem with the parameters $(k_A - k_B, k_B, p, l + p - f*)$ where $f* = \min_{w \in L(k_A, k_B)} F(p, l, w)$

Proof:

If s is a {0,1}-sequence with the [10]-property, then we can construct $S = z(s)$ a sequence in $0, 1^{k_A - k_B}$ by removing all the 0 that follow a 1 in s. $S \in L(k_A - k_B, k_B)$. Note that since s has the [10]-property, z^{-1} exists and $z^{-1}(S) = s$ If e is a vector in $E(p, l)$, we construct $E \in E(p, l + p - e.s)$ by the following procedure, E is equal to e but all the component that have been removed to go from s to $S = z(s)$ are also removed from e. If $e = e_1 + e_2 + .. + e_p$ where the e_i are all intervals, $l = l_1 + \cdots + l_p$ where $l_i = |e_i|$ and $e.s = f_1 + f_2 + \cdots + f_p = e_1.s + e_2.s + \cdots + e_p.s$ we also have $E = E_1, \cdots, E_p$ intervals, $L = L_1 + L_2 + \cdots + L_p$ and $E.S = F_1 + \cdots + F_p$. We have $L_i = l_i + 1 - f_i$ and $F_i = f_i$. So finally, $L = l + p - e.s$ and $E.S = e.s$.

Let s be the minimal sequence for k_A. Let $S = z(s)$. Capital letters will always refer the variables in $L(k_A - k_B, k_B)$. and small letters to the original problem (in $L(k_A, k_B)$).

We denote $f^* = \min\{F(p, l, s) : s \in L(k_A, k_B)\}$ and $F^* = \min\{F(p, l + p - f^*, S) : S \in L(k_A - k_B, k_B)\}$

If s is the minimal sequence we have $F(p, l, s) = f^*$ this implies that $F(p, L = l + p - f^*, S) = F$ with $F \geq f^*$. But this implies $F(p, l' = L - p + F, s) = f'$ with $f' \geq F$ which in turn implies $F(p, L' = l' + p - f', S) = F'$ with $F' \geq f'$. Now we got, $F' \geq F$ while $L' \leq L$. This means that $F' = F$ and therefore, $f' = F$. Finally we get $l' = l - f^* + F$ and $F(p, l - f^* + F, s) = F$ where $F \geq f^*$. In other words, this means that adding $F - f^*$ elements to some $e \in E_{p,l}$ which is an argmax for s, we increase $e.s$ by $F - f^*$. Therefore all this added elements are 1's in s. But as e is an argmax, all its intervals begin and end with a one (in s). So the [10]-property makes this requirement impossible. Therefore, $F = f^*$ and therefore $F^* \leq f^*$.

On the other hand, pick S to be minimal and let s be its image by z^{-1}. we get $F(, p, L = l + p - f^*, S) = F^* \rightarrow F(p, l = L - p + F^*, s) = f \rightarrow F(p, L' = l + p - f, S) = F'$ with $F' \geq f \geq F^*$. But then, $L' \leq L$ which implies that $F' = F^*$, which means $F(p, l' = l + F^* - f^*, s) = F^*$.

Since $F^* \leq f^*$, then $F(p, l, s) \leq f^*$ so s is minimal, $F(p, l, s) = f^*$. But now this implies $F(p, L, S) = F$ with $F \geq f^* \geq F^*$ (see the first part of this proof). As we have $F(p, L, S) = F*$ to start with, we can conclude $F = F^*$ and therefore $f^* = f$.

Finally we get that s is minimal if and only if $z(s)$ is minimal and their respective argmax are the transform from one another. ∎

8 This is the last part of the proof of the lemma (!) Since Theorem 1 is true for $l = p$ and since $p - f < 0$ if $l > p, k_B > p$, an induction on l will end the proof.

First note that if $\min_{w \in L(k_A,k_B)} F(p,l,w) > \min_{w \in L(k_A,k_B)} F(p,l-1,w)$
then $\min_{w \in L(k_A-k_B,k_B)} F(p,l+p-f^*,w) > \min_{w \in L(k_A-k_B,k_B)} F(p,l+p-f^*-1,w)$.
Indeed, if $\min_{w \in L(k_A-k_B,k_B)} F(p,l+p-f^*,w) = \min_{w \in L(k_A-k_B,k_B)} F(p,l+p-f^*-1,w)$,
then there exists one argmin of any minimal sequence S that contains a zero which implies that the corresponding argmin of s also contains a zero, and since all the minimal sequences in the original problem are transforms from minimal sequence in the transformed problem, this contradicts the item 7.

Now by the induction assymption, we can suppose the lemma is true for $F(p,l+p-f^*,.)$, that is for all $W \in L(k_A-k_B,k_B)$, if $F(p,l+p-f^*,W) \geq F(p,l+p-f^*,R)$ where R is the most regular sequence in $L(k_A-k_B,k_B)$.

If r is the most regular sequence in $L(k_A,k_B)$, Note that $z(r) = R$ (see property 6). Now we apply the result of item 5.2 to claim that for any sequence w, $F(p,l,w) \geq F(p,l,r)$. ■

Proof of Theorem 1

If If (p_r, l_r) verifies condition (1),

$$\min_{w \in L(k_A,k_B)} F(p_r,l_r,w) > \min_{w \in L(k_A,k_B)} F(p_r,l_r,w)$$

then

lemma 1 can be used. Let s be an arbitrary sequence. We get $t(C_r) \leq t(C)$ for any circuit C in MG(S,s) with p_r jumps and l_r places in C_B. Since $t(C) \leq t(C_s)$, we can conclude $t(C_r) \leq t(C_s)$.

If

$$\min_{w \in L(k_A,k_B)} F(p_r,l_r,w) = \min_{w \in L(k_A,k_B)} F(p_r,l_r-1,w)$$

then, there exists i such that $\min_{w \in L(k_A,k_B)} F(p_r,l_r-i,w) = F(p_r,l_r-i,r)$
either because $\min_{w \in L(k_A,k_B)} F(p_r,l_r-i,w) > \min_{w \in L(k_A,k_B)} F(p_r,l_r-i-1,w)$ and using

lemma 1
or because $l_r - i = p_r$.

Let s be an arbitrary sequence. We have

$$\begin{aligned}
t(C_s) &\geq F(p_r, l_r - i, s)(b_R + b_{\bar{R}}) + (k_A - l_r + i)(a_R + a_{\bar{R}}) + p_r(b_{\bar{R}} - a_R) \text{ [weight of a circuit]} \\
&\geq F(p_r, l_r - i, r)(b_R + b_{\bar{R}}) + (k_A - l_r)(a_R + a_{\bar{R}}) + p_r(b_{\bar{R}} - a_R) + i(a_R + a_{\bar{R}}) \text{ [lemma 1]} \\
&\geq (F(p_r, l_r - i, r) + i)(b_R + b_{\bar{R}}) + (k_A - l_r)(a_R + a_{\bar{R}}) + p_r(b_{\bar{R}} - a_R) \quad [a_R + a_{\bar{R}} \geq b_R + b_{\bar{R}}] \\
&\geq F(p_r, l_r, r)(b_R + b_{\bar{R}}) + (k_A - l_r)(a_R + a_{\bar{R}}) + p_r(b_{\bar{R}} - a_R) \text{ [} r \text{ is a } \{0,1\}\text{-sequence]} \\
&= t(C_r)
\end{aligned}$$

This ends the proof of Theorem 1. ■

6 N Processes Sharing a Resource

In the case of N processes sharing a resource with $N \geq 3$, Theorem 1 cannot be extended. Indeed, in general, no strongly optimal allocation sequence exists. The optimal sequence depends on the temporizations of the processes.

To illustrate this negative result, we give an example where the frequencies of the $n = 3$ processors are fixed but if we change the temporizations, then the optimal sequence changes as well.

During $k = 6$ allocations, the resource must be given $k_A = 3$ times to process A, $k_B = 2$ times to process B and $k_C = 1$ times to process C.

With the temporizations: $(1, 1)$ for A, $(1, 1)$ for B and $(1, 1)$ for C, then the optimal sequence is *cababa* (and all its shifts of course). This sequence gives a cycle time of 6. But with the temporizations: $(1, 1)$ for A, $(1, 3)$ for B and $(2, 1)$ for C, then the optimal sequence is *cbaaba* (and all its shifts).

This negative result does not put to an end the search for an optimal sequence in the case where we have more than two processes sharing a resource. Although finding the optimal allocation of a token in a general Petri Net is an NP-Hard problem, see [Ramamoorthy 80], we believe that a "even distribution" of the heaviest processors in terms of (*processing time*) *(*frequency*) could help to solve the problem or at least is a good heuristic to find a good allocation.

7 Generalized Optimal Sequences

In the previous sections we have been interested only in periodic sequences. Here, we show that the results obtained in the periodic case can be extended to more general cases.

7.1 Non Periodic Sequences

We denote by $p_m(w)$ the prefix of w of length m. $V(k_A, k_B) = \{w, \forall n \in IN\ |p_{nk}(w)|_A = n.k_A,\ |p_{nk}(w)|_B = n.k_B\}$ We also denote $T_m(w)$ the time to go through the first m allocations of w. A general sequence $w \in V(k_A, k_B)$ is not periodic in general (of any period).

Theorem 2 *$\exists a_0$ s.t. $\forall a \geq a_0$, $\forall w \in V(k_A, k_B)$, $T_{nk}(w) \geq T_{nk}(r^\omega)$.*

Proof: $T_{nk}(w) = T_{nk}(p_{nk}(w)^\omega)$ because $T_{nk}(w)$ depends only on $p_{nk}(w)$. But now, $p_{nk}(w)^\omega$ is periodic of period nk. We get: $T_{nk}(p_{nk}(w)^\omega) \geq T_{nk}(r^\omega)$ if $n \geq n_0$ for n_0 large enough, since the cycle time of the system $MG(S, T, p_{nk}(w)^\omega)$ is bigger than the cycle time of $MG(S, T, r\omega)$. ■

7.2 Asymptotic Constraints

An even more general set of sequences would be sequences with asymptotic constraints only. Let $A(k_A, k_B)$ be the set of the infinite sequences w verifying: $lim_{m\to\infty} \frac{|p_m(w)|_A}{m} = \frac{k_A}{k}$ and $lim_{m\to\infty} \frac{|p_m(w)|_B}{m} = \frac{k_B}{k}$. In this case, we can state an asymptotic inequality:

Theorem 3 *For all $w \in A(k_A, k_B)$, $T_{nk}(w) \geq T_{nk}(r^\omega) + o(n)$.*

Remark : if the limits exist, the previous theorem implies that $\lim_{n\to\infty} \frac{T_{nk}(w)}{nk} \geq \lim_{n\to\infty} \frac{T_{nk}(r^\omega)}{nk}$.

Proof: Let n be an arbitrary integer. $|p_{nk}(w)|_A = nk_A + \epsilon_A$ and $|p_{nk}(w)|_B = nk_B + \epsilon_B$ where $\varepsilon_A = o(n)$ and $\varepsilon_B = o(n)$.

If n is big enough,

$$\begin{aligned} T_{nk}(p_{nk}(w)) &\geq T_{nk}(r^n) - |\varepsilon_A|(\alpha_1 + \alpha_2) - |\varepsilon_B|(\beta_1 + \beta_2) \\ &= nT_k(r) + o(n). \end{aligned}$$

7.3 Irrational Frequencies

Suppose now that we impose irrational frequencies for the allocation of the resource. Namely, we consider the set $I(x)$ of all the sequences w verifying $lim_{m\to\infty}\frac{|p_m(w)|_A}{m} = x$ where x is irrational, $0 < x < 1$. In this case we cannot define the most regular sequence in the sense we used so far. However, there exists a similar notion. If x is any real number we can define the *word of Sturm*, $r(x)$ (first introduced in [Morse 40]) associated with x. If x is rational ($x = a/b$) this infinite word coincides with $r(a,b)^\omega$, the most regular word associated with the couple (a,b). Indeed one possible definition of $r(x)$ is: $r(x) = \lfloor i.x \rfloor = \lfloor (i-1)x \rfloor$

Theorem 4 *For all $w \in I(x)$, $T_n(w) \geq T_n(r(x)) + o(n)$.*

Proof: We build a sequence of rational numbers p_n, $\lim_{n\to\infty} p_n = x$ such that for all $i \leq n$, $\lfloor i.p_n \rfloor = \lfloor i.x \rfloor$. The sequence (p_n) is defined such that r_n, the most regular word with the frequency p_n and $r(x)$, the word of Sturm associated with x have the same n first letters. Now, if w is a word in $I(x)$, $|p_n(w)|_A = |r_n|_A + \eta_A(n)$, $|p_n(w)|_B = |r_n|_B + \eta_B(n)$, where
$lim_{n\to\infty}\eta_A(n) = 0$, $\lim_{n\to\infty} \eta_B(n) = 0$. Now ,

$$\begin{aligned} T_n(w) &\geq T_n(r_n) - |\eta_A + 1|(\alpha_1 + \alpha_2) - |\eta_B + 1|(\beta_1 + \beta_2) \\ &= T_n(r_n) + o(n) \\ &= T_n(r(x)) + o(n). \end{aligned}$$

8 conclusion

We have presented one scheduling application of the most regular words with fixed letter frequencies to the optimization of a scheduling scheme of a discrete event system. We believe that many other applications of this interesting notion can be investigated in various domains.

References

[Chaouiya 92] Chaouiya, C., 1992. Outils pour l'analyse de systèmes synchronis'es. PhD thesis, University of Nice Sophia Antipolis, France.

[Baccelli 92] Baccelli, F., Cohen, G., Oldser, G.J., Quadrat, J.-P., 1992. *Synchronization and Linearity.* Springer Verlag, New-York.

[Berstel 90] Berstel, j., 1990. Tracé de droites, fractions continues et morphismes itérés. *Mots, M. Lothaire (Ed)*, Hermès, Paris, 298-309.

[Hajek 85] Hajek, B., 1985. Extremal splittings of point processes. *Mathematics of Operation Research*, 10:543-556.

[Gaujal 93] Gaujal, B., Jafari, M., Gursöy-Baykal, M., Alpan, G., 1993. Allocation sequence of two processes sharing a ressource. *Submitted.*

[Morse 40] Morse, M., Hedlund, G.A., 1940. Symbolic dynamics II - Sturmian trajectories. *Amer. J. Math.* 62,1-42.

[Murata 84] Murata, T., 1984. Petri Nets: Properties, analysis and applications. *Proceeedings of IEEE*, 77-4.

[Ramamoorthy 80] Ramamoorthy C.V., Ho, G.S., 1980. Performance Evaluation os Asynchronous Concurrent Systems Using Petri Nets. *IEEE trans. Software Eng.*, SE-6:440-449.

[Rauzy 84] Rauzy, G., 1984. Mots infinis en arithmétique. *Automata of infinite words, Nivat, Perrin (Eds)* LNCS 192, Springer-Verlag, 165-171.

Unité de recherche INRIA Lorraine, Technôpole de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399